

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Алгоритмы и структуры данных»
Тема: Структуры данных: стек, очередь.

Студент гр. 8304

Бутко А.М.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2019

Цель работы.

Изучить и реализовать стек через динамическую (связанную) память (на базе указателей).

Задание.

Вариант 4-д.

Содержимое заданного текстового файла F, разделенного на строки, переписать в текстовый файл G, выписывая литеры каждой строки в обратном порядке.

Выполнение работы.

В заголовочном файле Stack.h содержится класс стека на основе динамической памяти Stack. В приватном поле класса содержится структура node_, в которой содержится указатель на следующий элемент и сам элемент, а также указатель на элемент на вершине стека. Методы работы со стеком находятся в открытой части класса.

Описание методов:

1. Stack() и ~Stack() - конструктор и деструктор соответственно.
2. push() - кладет элемент на вершину стека.
3. isEmpty() - проверка стека на пустоту.
4. pop() - возвращает элемент с вершины стека и удаляет его.

Описание работы программы.

В main() происходит открытие файла с текстом и файла, куда будут записаны результаты, объявление стека, вызов функции reverseFunction(). В данной функции файл построчно считывается, литеры строки кладутся по очереди на вершину стека, затем так же по очереди снимаются с вершины стека и записываются в файл с результатами.

Тестирование.

Input:	Output:
<p>Reach out and touch faith</p> <p>Your own personal Jesus</p> <p>Someone to hear your prayers Someone who cares</p> <p>Your own personal Jesus</p> <p>Someone to hear your prayers Someone who's there Feeling unknown And you're all alone Flesh and bone By the telephone Lift up the receiver I'll make you a believer Take second best Put me to the test Things on your chest You need to confess</p>	<p>htiaf hcuot dna tuo hcaeR</p> <p>suseJ lanosrep nwo ruoY</p> <p>sreyarp ruoy raeh ot enoemoS serac ohw enoemoS</p> <p>suseJ lanosrep nwo ruoY</p> <p>sreyarp ruoy raeh ot enoemoS ereht s'ohw enoemoS nwonknu gnileeF enola lla er'uoy dnA enob dna hself enohpelet eht yB reviecer eht pu tfiL reveileb a uoy ekam ll'I tseb dnoceS ekaT tset eht ot em tuP tsehc ruoy no sgnihT ssefnoc ot deen uoY</p>
<p>htiaf hcuot dna tuo hcaeR</p> <p>suseJ lanosrep nwo ruoY</p> <p>sreyarp ruoy raeh ot enoemoS serac ohw enoemoS</p> <p>suseJ lanosrep nwo ruoY</p> <p>sreyarp ruoy raeh ot enoemoS ereht s'ohw enoemoS nwonknu gnileeF enola lla er'uoy dnA enob dna hself enohpelet eht yB reviecer eht pu tfiL</p>	<p>Reach out and touch faith</p> <p>Your own personal Jesus</p> <p>Someone to hear your prayers Someone who cares</p> <p>Your own personal Jesus</p> <p>Someone to hear your prayers Someone who's there Feeling unknown And you're all alone Flesh and bone By the telephone Lift up the receiver</p>

reveileb a uoy ekam ll'I tseb dnoceS ekaT tset eht ot em tuP tsehc ruoy no sgnihT ssefnoc ot deen uoY	I'll make you a believer Take second best Put me to the test Things on your chest You need to confess
A B C D E F G	A B C D E F G

Выводы.

Был реализован класс на основе структуры данных по принципу LIFO стека на базе указателей. Были изучены некоторые принципы Google C++ Style Guid.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл main.cpp:

```
#include "Stack.h"

#include <iostream>
#include <fstream>

void reverseFunction(std::ifstream& file, std::ofstream& newFile,
Stack& aStack)
{
    std::string line;
    while (getline(file, line))
    {
        aStack.push('\n');
        for (char i : line)
            aStack.push(i);
        while (!aStack.isEmpty())
            newFile << aStack.pop();
        aStack.pop();
    }
}

int main()
{
    Stack aStack;
    std::string fileName, newFileName;
    std::cout << "Enter test-file location:" << std::endl;
    std::cin >> fileName;
    std::ifstream file;
    file.open(fileName);
    if (!file.is_open())
    {
        std::cout << "error: file is not open" << std::endl;
        return 0;
    }

    std::cout << "Enter where to save results (location with
<name>.txt): " << std::endl;
    std::cin >> newFileName;
    std::ofstream newFile(newFileName);
    if (!newFile.is_open())
    {
        std::cout << "error: file is not open" << std::endl;
        return 0;
    }
    reverseFunction(file, newFile, aStack);
    return 0;
}
```

Файл Stack.h:

```
#pragma once

class Stack
{
public:
    Stack():top(nullptr) {}
    void push(char element)
    {
        auto* tmp = new node_;
        tmp->element_ = element;
        tmp->prev_ = top;
        top = tmp;
    }
    bool isEmpty()
    {
        return top == nullptr;
    }
    char pop()
    {
        if (isEmpty()) return -1;
        char element = top->element_;
        auto* tmp = top;
        top = top->prev_;
        delete tmp;
        return element;
    }
    ~Stack()
    {
        while (top)
        {
            auto* tmp = top;
            top = top->prev_;
            delete tmp;
        }
    }

private:
    struct node_
    {
        char element_;
        node_* prev_;
    };
    node_* top;
};
```